

#Include 자료구조 JAVA - A

1. 네 값의 최댓값을 구하는 max4 메서드를 작성하세요. 작성한 메서드를 테스트 하기 위해 main 메서드를 포함한 프로그램을 작성해야 합니다. 이후의 문제도 마찬가지입니다.

```
package chapter1;

public class Q1 {

    static int max4(int a,int b,int c,int d) {
        int max = a;
        if(max < b)
            max = b;
        if(max < c)
            max = c;
        if(max < d)
            max = d;

        return max;
    }

    public static void main(String[] args) {
        System.out.println("max4(3,1,2,4) = " +
max4(3,1,2,4));
    }

}
```

2. 세 값의 최솟값을 구하는 min3 메서드를 작성하세요.

```
package chapter1;

public class Q2 {
    static int min3(int a, int b, int c) {
        int min = a;
        if(b<min) min = b;
        if(c<min) min = c;

        return min;
    }

    public static void main(String[] args) {
        System.out.println("min3(3,2,1) = " + min3(3,2,1));
        System.out.println("min3(1,2,3) = " + min3(3,2,1));
        System.out.println("min3(2,3,1) = " + min3(3,2,1));
    }

}
```

3. 네 값의 최솟값을 구하는 min4 메서드를 작성하세요.

```
package chapter1;
```

```
public class Q3 {  
    static int min4(int a, int b, int c, int d) {  
        int min = a;  
        if(b<min) min = b;  
        if(c<min) min = c;  
        if(d<min) min = d;  
  
        return min;  
    }  
  
    public static void main(String[] args) {  
        System.out.println("min4(3,2,1,4) = " +  
min4(3,2,1,4));  
        System.out.println("min4(1,4,2,3) = " +  
min4(1,4,2,3));  
        System.out.println("min4(2,3,4,5) = " +  
min4(2,3,4,5));  
    }  
}
```

4. 세 값의 대소 관계인 13가지 조합의 중앙값을 구하여 출력하는 프로그램을 작성하세요.

```
package chapter1;
```

```
public class Q4 {  
    public static int med3(int a, int b, int c) {  
        if(a>=b)  
            if(b>=c)  
                return b;  
            else if(a<=c)  
                return a;  
            else  
                return c;  
        else if (a>c)  
            return a;  
        else if(b>c)  
            return c;  
        else  
            return b;  
    }  
}
```

5. 중앙값을 구하는 메서드는 다음과 같이 작성할 수도 있습니다. 그러나 실습 1C-1의 med3 메서드에 비해 효율이 떨어지는데 그 이유를 설명하세요.

<코드>

```
package chapter1;

public class Q5 {
    static int med3(int a, int b, int c) {
        if((b>=a && c<=a) || (b <=a && c>= a))
            return a;
        else if ((a>b&& c< b) || (a<b && c> b))
            return b;
        else return c;
    }
}
```

<코드 : 4번과 5번의 연산 시간 차이를 구하는 코드>

```
package chapter1;
import java.time.Duration;
import java.time.Instant;
public class Q4_5 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        // 첫 번째 코드 실행 시간 측정
        Instant start1 = Instant.now();
        Q4.med3(1, 2, 3);
        Instant end1 = Instant.now();
        Duration duration1 = Duration.between(start1, end1);
        System.out.println("첫 번째 코드 실행 시간: " + duration1.toNanos() + "ns");

        // 두 번째 코드 실행 시간 측정
        Instant start2 = Instant.now();
        Q5.med3(1, 2, 3);
        Instant end2 = Instant.now();
        Duration duration2 = Duration.between(start2, end2);
        System.out.println("두 번째 코드 실행 시간: " + duration2.toNanos() + "ns");
    }
}
```

4번 코드 실행 시간 : 30200ns

5번 코드 실행 시간 : 19800ns

4번 코드는 해당되는 조건의 연산을 수행하다보면 최종 결과를 일찍 알 수 있는 순서가 있는데에 반해, 5번 코드는 순서도가 없이 결과를 구해도 연산이 수행되는 경우가 있어

비효율적인 코드이다.

6. 실습 1-7에서 while문이 종료될 때 변수 i값이 n+1이 되는 지 확인하세요

```
package chapter1;

import java.util.Scanner;

public class Q6 {

    public static void main(String[] args) {
        Scanner stdIn = new Scanner(System.in);

        System.out.println("1부터 n까지의 합을 구합니다.");
        System.out.println("n 값 : ");
        int n = stdIn.nextInt();
        stdIn.close();
        int sum = 0;
        int i = 1;

        while( i <= n) {
            sum += i;
            i++;
        }

        System.out.println("1부터 " + n + "까지의 합 : " +
sum);
        System.out.println("i 값 : " + i);
    }
}
```

7. 1~10의 합은 $(1+10) * 5$ 와 같이 구할 수 있습니다. 이를 ‘가우스의 덧셈’이라고 하는데 이 방법을 이용하여 1부터 n까지의 정수 합을 구하는 프로그램을 작성하세요

```
package chapter1;

import java.util.Scanner;

public class Q7 {
    static int gaus(int n) {
        int sum = 0;
        if(n%2 == 0) {
            for(int i=1 ; i<=(n/2); i++) {
                sum += 1+n;
            }
        }
    }
}
```

```

        System.out.println(sum);
    }
}
else {
    for(int i=1 ; i<=(n/2); i++) {
        sum += 1+n;

        System.out.println(sum);
    }
    sum += ((int) n/2)+1;

    System.out.println(sum);
}
return sum;
}

public static void main(String[] args) {
    //가우스 덧셈 코드 짜기

    Scanner stdIn = new Scanner(System.in);

    System.out.println("1부터 n까지의 합을 구합니다.");
    System.out.println("n 값 : ");
    int n = stdIn.nextInt();

    System.out.println("1부터 " + n + "까지의 합 : " +
gaus(n));

    stdIn.close();
}
}

```

8.정수 a,b를 포함하여 그 사이의 모든 정수의 합을 구하여 반환하는 메서드를 작성하세요.

```

package chapter1;

import java.util.Scanner;

public class Q8 {
    static int sumof(int a, int b) {
        int sum= 0;

        for(int i=a; i<=b; i++) {
            sum += i;
        }
    }
}

```

```

        return sum;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Scanner stdIn = new Scanner(System.in);

        System.out.println("a부터 b까지의 합을 구합니다.");
        System.out.println("첫 번째 값 : ");
        int a = stdIn.nextInt();

        System.out.println("두 번째 값 : ");
        int b = stdIn.nextInt();

        System.out.println(a + "부터 " + b + "까지의 합 : " +
sumof(a, b));

        stdIn.close();
    }
}

```

9.오른쪽 결과와 같이 두 변수 a,b에 정수를 입력하고 b-a를 출력하는 프로그램을 작성하세요.

```

package chapter1;

import java.util.Scanner;

public class Q9 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner stdIn= new Scanner(System.in);
        System.out.println("-- b-a 출력 프로그램 --");
        System.out.println("a 값 : ");
        int a = stdIn.nextInt();
        int b;
        do{
            System.out.println("b 값 : ");
            b = stdIn.nextInt();
            if(b<a)
                System.out.println("a 보다 크거나 같은 값을 입력
하세요. ");
        }while(b<a);
    }
}

```

```

        System.out.println("b - a = " + (b-a));

        stdIn.close();
    }
}

```

10. 양의 정수를 입력하고 자릿수를 출력하는 프로그램을 작성하세요. 예를 들어 135를 입력하면 ‘그 수는 3자리입니다’라고 출력하고, 1314를 입력하면 ‘그 수는 4자리입니다’라고 출력합니다.

```

package chapter1;

import java.util.Scanner;

public class Q10 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner stdIn = new Scanner(System.in);

        long a;
        do {
            System.out.println("자릿수를 알고 싶은 양의 정수를 입력
하세요 :");

            a = stdIn.nextInt();
            if(a<=0)
                System.out.println("0 보다 큰 정수를 입력하세
요.");
        }while(a<=0);

        long digits = (long) Math.log10(a) + 1;

        System.out.println("그 수는 " + digits + " 자리 입니
다.");

        stdIn.close();
    }
}

```

11.오른쪽 결과와 같이 위쪽과 왼쪽에 곱하는 수가 있는 구구단 곱셈표를 출력하는 프로그램을 작성하세요.

```
package chapter1;
```

```
public class Q11 {
    public static void main(String[] args) {
        System.out.printf("%2s |", "");
        for(int i = 1; i<=9; i++) {
            System.out.printf("%3d", i);
        }
        System.out.println();
        System.out.println("----+-----");
        for(int i = 1; i <= 9; i++) {
            System.out.printf("%2d |", i);
            for(int j =1; j <= 9; j++)
                System.out.printf("%3d", i*j);
            System.out.println();
        }
    }
}
```

12.구구단 곱셈표를 변형하여 곱셈이 아니라 덧셈을 출력하는 프로그램을 작성하세요

```
package chapter1;
```

```
public class Q12 {
    public static void main(String[] args) {
        System.out.printf("%2s |", "");
        for(int i = 1; i<=9; i++) {
            System.out.printf("%3d", i);
        }
        System.out.println();
        System.out.println("----+-----");
        for(int i = 1; i <= 9; i++) {
            System.out.printf("%2d |", i);
            for(int j =1; j <= 9; j++)
                System.out.printf("%3d", i+j);
            System.out.println();
        }
    }
}
```



```
}
```

13.오른쪽 결과와 같이 입력한 수를 한 번으로 하는 정사각형을 *로 출력하는 프로그램을 작성하세요

```
package chapter1;

import java.util.Scanner;

public class Q13 {
    public static void main(String[] args) {
        System.out.println("정사각형을 출력합니다.");
        Scanner stdIn = new Scanner(System.in);
        int a = stdIn.nextInt();

        for(int i=1; i<=a;i++) {
            for(int j=1; j<=a; j++) {
                System.out.print("*");
            }
            System.out.println();
        }

        stdIn.close();
    }
}
```

14.직각이등변삼각형을 출력하는 부분을 아래와 같은 형식의 메서드로 작성하세요

또, 왼쪽 위, 오른쪽 위, 오른쪽 아래가 직각인 이등변 삼각형을 출력하는 메서드를 각각 작성하세요

```
package chapter1;

import java.util.Scanner;

public class Q14 {
    static void triangleLB(int n) {
        for(int i=1; i<=n;i++) {
            for(int j=1; j<=i; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

```

    }
}

static void triangleLU(int n) {
    for(int i=1; i<=n; i++) {
        for(int j=n; j>=i; j--) {
            System.out.print("*");
        }
        System.out.println();
    }
}

static void triangleRU(int n) {
    for(int i=1; i<=n; i++) {
        for(int j=1; j<=n; j++) {
            if(j>=i)
                System.out.print("*");
            else
                System.out.printf("%1s", "");
        }
        System.out.println();
    }
}

static void triangleRB(int n) {
    for(int i=1; i<=n; i++) {
        for(int j=1; j<=n; j++) {
            if(j>=n+1-i)
                System.out.print("*");
            else
                System.out.printf("%1s", "");
        }
        System.out.println();
    }
}

public static void main(String[] args) {
    System.out.println("직각 이등변 삼각형을 출력합니다.");
    System.out.println("몇 단 삼각형입니까?");
    Scanner stdIn = new Scanner(System.in);
    int a = stdIn.nextInt();

    triangleLU(a);
    triangleLB(a);
    triangleRU(a);
    triangleRB(a);

    stdIn.close();
}
}

```

15.n단의 피라미드를 출력하는 메서드를 작성하세요

```
package chapter1;

import java.util.Scanner;

public class Q15 {
    static void spira(int a) {
        for(int i=1; i<=a;i++) {
            for(int j=1; j<=(a*2)-1; j++) {
                if(j<=a+(i-1) && j>=a-(i-1))
                    System.out.print("*");
                else
                    System.out.printf("%1s", "");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        System.out.println("피라미드 출력합니다.");
        Scanner stdIn = new Scanner(System.in);
        int a = stdIn.nextInt();

        spira(a);

        stdIn.close();
    }
}
```

이 부분은 팀원과 비교해 보았을 때, 다른 팀원보다 비효율적인 코드로 작성되었다. 이유는

내 코드는 일부러 출력되는 범위를 항상 $(a*2)-1$ 만큼으로 설정하고, 그 안에서 *을 출력하고 싶은 범위를 설정하는 식이었는데, 팀원들은 필요한만큼의 공백과 *만 출력하여 출력 결과를 더 효율적으로 작성하였기 때문이다.

16.오른쪽과 같이 아래를 향한 n단의 숫자 피라미드를 출력하는 메서드를 작성하세요

```
package chapter1;

import java.util.Scanner;

public class Q16 {
    static void spira(int a) {
```

```

        for(int i=1; i<=a;i++) {
            for(int j=1; j<=(a*2)-1; j++) {
                if(j<=a+(i-1) && j>=a-(i-1))
                    System.out.printf("%d", i%10);
                else
                    System.out.printf("%1s", "");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        System.out.println("피라미드 출력합니다.");
        Scanner stdIn = new Scanner(System.in);
        int a = stdIn.nextInt();

        spira(a);

        stdIn.close();
    }
}

```